Sparse online learning with bandit feedback

Hongliang Zhong, Emmanuel Daucé

Sparse online learning with bandit feedback

Hongliang Zhong[a], Emmanuel Daucé[b,*]

[a] *Aix Marseille Univ, CNRS, Centrale Marseille, LIF, Laboratoire d'Informatique Fondamentale, Marseille, France*
[b] *Aix Marseille Univ, Inserm, INS, Institut de Neurosciences des Systèmes, Marseille, France*

Abstract

The bandit classification problem considers learning the labels of a time-indexed data stream under a mere "hit-or-miss" binary guiding. Adapting the OVA ("one-versus-all") hinge loss setup, we develop a sparse and lightweight solution to this problem. The issued sequential norm-minimal update solves the classification problem in finite time in the separable case, provided enough redundancy is present in the data. An $O(\sqrt{T})$ regret is moreover expected in the non-separable case. The algorithm shows effectiveness on both large scale text-mining and machine learning datasets, with (i) a favorable comparison with the more demanding confidence-based second-order bandits setups on large scale datasets and (ii) a good sparsity and efficacy when a kernel approach is applied to non-separable datasets.

Keywords: Contextual bandits, Hinge loss, Online learning, Kernel methods

1. Introduction

Categorical online learning is a well-documented family of learning problems in which (i) a time order is defined on the sequence of input examples x_1, ..., x_t, ... with the classifier update taking place after each example presentation and (ii) the output space is categorical, i.e. a single response y_t, among $K > 1$ possibilities, is expected at trial t. The multiclass learning task typically addresses object recognition (such as OCR, face recognition,

*Corresponding author

speech recognition etc.) and recommender systems. In contrast with the offline approach, online learning takes the form of an iterative process, relying on a time-ordered sequence of observations, actions and feedbacks (where the feedback is the observable outcome of the action). Two well-known setups obey to this class, i.e. the contextual bandit setup (see Lai and Robbins (1985) and Auer et al. (2002)) and supervised online learning setup (see Rosenblatt (1958), Duda et al. (1973) and Freund and Schapire (1999)), the two setups mainly differing by the nature of the feedback. Whether abundant or scarce, deterministic or stochastic, stationary or adversarial, the feedback characteristics critically shape the design of the learning algorithms.

The case we address in the following is the one-bit feedback in multiclass classification tasks, occasionally called the bandit feedback classification problem (see Kakade et al. (2008)), for the outcome is either 0 (failure) or 1 (success). This case lies at the crossroad of the supervised online learning setup and the bandit setup. A one-bit feedback reflects a "hit-or-miss" learning situations, in which a single bit indicates to the learner whether its categorical response was correct ("hit") or incorrect ("miss"). Wether simple in its principle, this problem is bound to the online approach and was only recently addressed by the community (see Kakade et al. (2008), Crammer and Gentile (2013), etc.). Apart from the original work of Kakade et al. (2008), most recent approaches to this problem are based on regularized linear models (see Li et al. (2010), Hazan and Kale (2011), Crammer and Gentile (2013), Ngo et al. (2013)), to which the powerful methods of contextual bandit, including UCB-like non-stochastic exploration strategy (see Lai and Robbins (1985)), do apply. While providing almost optimal convergence rates, online multivariate setups suffer from a quadratic complexity in space that limits their applicability to large-dimensional datasets.

In contrast, the sequential approach to quadratic optimization, as proposed by Anlauf and Biehl (1989) and Crammer et al. (2006), may both provide upper bounds on error rate and a smaller memory footprint, i.e. only display linear scaling in space. We adapt in this paper a similar view to the bandit feedback case, where online learning relies on a local norm minimization under standard linear-convex constraints. Our conservative approach allows to determine similar regret bounds than the original paper of Crammer et al. (2006), providing strong guaranties on convergence capabilities,

that are confronted to other methods over synthetic and real datasets[1].

2. Outlook

2.1. Online learning

Online learning is a *process* in which an agent (the "learner") probes its environment sequentially to obtain *information* that is eventually used to improve its *fitness*. The universe being initially hidden, the online approach works on probing the database (universe) through individual queries that step-by-step unveil the environment.

The sequential organization of learning is present in the most traditional setups such as the Bandit problems (see Robbins (1952)) and in the Perceptron algorithm (see Rosenblatt (1958)). The general setup we consider here is the case of an "open-loop" *actionnable* universe. The time-indexed observations x_1, ..., x_t, ... are independent (causally disconnected). Each observation causes the learner to output a single action $\tilde{y}_t$ out of $K > 1$ possible actions. Every action provides a feedback f_t that is an *information* about a *value of interest* (or reward) r_t that needs to be maximized over time. The feedback is *explicit* in most contextual bandit setups, i.e. $f_t = r_t$ (see Lai and Robbins (1985) and Auer et al. (2002)) while it takes the form of a category $f_t = y_t$ in traditional online learning setups, that *indirectly* provides a quantity to maximise through a loss function $l(x_t, \tilde{y}_t, f_t) = -r(x_t, \tilde{y}_t, f_t)$ (see Duda et al. (1973), Freund and Schapire (1999), Kivinen et al. (2004), Crammer et al. (2006)). Solving the problem thus means both learning the universe from experience and optimizing the final reward.

2.2. Linear models

Multiclass classification requires finding an appropriate class $y \in \{1, ...K\}$ for every observation vector $x \in \mathbb{R}^d$. The *decision function* (or policy) allows to carry out a categorical response $\tilde{y}$. The decision function embodies both prior information about the regularity of the universe and experience-based information. A reasonable assumption is to consider that close-by contexts

[1]This work was partly presented at the ECML-PKDD doctoral consortium (see Zhong and Daucé (2015)). Substantial refinements and new notations are used in the present paper. The kernel extension, with corresponding developments and simulations, were not present in the conference article.

should provide close-by rewards distributions. The linear approach to classification means define a set of linear mappings:

$$W = (w_1, .., w_K) \in \mathbb{R}^{Kd} \tag{1}$$

so that, for every observation x and every category k, a *similarity score* $\langle w_k, x \rangle$ is carried out. The linear assumption allows (i) to settle choice (or action) over class similarity prediction and (ii) to update models over prediction accuracy.

The set of linear mappings can either be considered class-prototypes or class-separatrices. The first case implements a *model-based* approach, and the second a *discriminant-based* approach. In the first case, second order gradient descent methods such as natural gradient (see Amari et al. (2000)), Gauss-Newton (see Le Cun and Bottou (2004)), and second order perceptron (see Cesa-Bianchi et al. (2005)) provide almost optimal convergence rates while scaling quadratically with observation vector dimension. In the second case, the quadratic optimization approach (see Anlauf and Biehl (1989), Crammer et al. (2006)) only provide upper bounds on the error rate but display linear scaling in size, label-unbalance robustness and more generally provide a smaller memory footprint.

2.3. One-bit bandit feedback

The one-bit bandit feedback offers a specific online learning setup that implements, in a principled way, the scarce labelling information problem. After reading $\tilde{y}_t$, instead of providing a label, the universe provides a single bit of information called the *label hit*, i.e. $f_t = 1$ if $\tilde{y}_t = y_t$ and $f_t = 0$ elsewhere. The objective, just as in the supervised case, is to avoid label misses and maximise labels hits. A specific update function needs to be defined to allow for label hit improvement over time.

Banditron. This problem was coined the "bandit feedback classification problem" by Kakade et al. (2008). Taking inspiration from the multiclass perceptron (Duda et al. (1973)), a time-effective algorithm, called the "Banditron", is established: K linear mappings $w_1, ..., w_K$ are defined and the candidate response $\hat{y}$ relies on a best-match policy, i.e.

$$\hat{y}_t = \underset{k \in \{1,..,K\}}{\operatorname{argmax}} \langle w_{k,t-1}, x_t \rangle \tag{2}$$

Contrarily to the supervised case, an *exploration* policy needs to be defined to efficiently sample the decision space. The Banditron adopts a simple $\varepsilon - greedy$ search allowing non-matching responses to be occasionally chosen in a uniform way:

$$P(\tilde{Y}_t = k) = (1 - \varepsilon)\delta(k, \hat{y}_t) + \frac{\varepsilon}{K} \tag{3}$$

with $\varepsilon \in [0, 1]$ and $\delta(i, j) = 1$ if $i = j$, and 0 elsewhere. Given a set of linear mappings W_{t-1} and an actual output $\tilde{y}_t$, the update at time t is :

$$W_t = W_{t-1} + \frac{\delta(\tilde{y}_t, y_t) X_t^{\tilde{y}_t}}{P(\tilde{Y}_t = \tilde{y}_t)} - X_t^{\hat{y}_t} \tag{4}$$

with the convention:

$$X_t^k \triangleq (\vec{0}, ..., x_t, ..., \vec{0}) \in \mathbb{R}^{Kd} \tag{5}$$
$$\Big|$$
$$k$$

a sequence of null vectors, except at position k where the observation vector x_t is found, so that $\langle W, X_t^k \rangle = \langle w_k, x_t \rangle$. The expectation of the update is shown to be that of the multiclass perceptron, and a convergence to the corresponding classifier is obtained.

Second-order models. Apart from Kakade et al. (2008), a majority of learning setups address the bandit classification problem using derivatives of the linUCB (see Auer (2002)) and/or second-order perceptron (see Cesa-Bianchi et al. (2005)). Those setups allow to carry out confidence intervals over reward predictions, based on the observation vectors covariance matrix $\langle {X_t^{\tilde{y}_t}}^\top X_t^{\tilde{y}_t} \rangle_t$, and adopt the UCB exploration policy (see Lai and Robbins (1985)), providing $O(\sqrt{T})$ regret bounds in the stationary case (see Li et al. (2010), Hazan and Kale (2011), Crammer and Gentile (2013), Ngo et al. (2013)). The close to optimal regret bounds obtained in that case are harmed by a quadratic space complexity and a lack of sparsity that justifies our closer inspection to the separatrix-based setup.

2.4. Online risk minimization

The computational efficiency of the perceptron (Rosenblatt (1958)), as well as the SVM (Vapnik (1998)), critically relies on their sparsity, i.e. their

capability to store only the most relevant observation vectors regarding classification task (the so-called "support vectors"). Fewer vectors in a classifier provide better generalization capabilities and participate in regularization. Under the discriminant approach to linear classification, a set of separating hyperplanes are expected to optimally separate the input space according to the misclassification risk. This risk can for instance be estimated through a margin principle (see Vapnik (1998)), imposing a non-zero distance of known class exemplars to the classification boundaries.

Following Kivinen et al. (2004) and Crammer et al. (2006), we consider a sequential approach to risk minimization, where classifiers are updated through local measures of a hinge loss function:

$$l_t = l(x_t, W_{t-1}, y_t)$$

Different hinge loss functions and corresponding margin constraints can be defined depending on the task and feedback characteristics. Under the multiclass setting, two principal margin constraint schemes can be set up, namely the *relative* margin and the *normative* margin.

- Compliant with the Kessler's construction (see Duda et al. (1973)), the *relative* margin setup (see Crammer and Singer (2003)) establish a distance reference a, so that the linear score of the class-compliant separatrix $\langle w_y, x\rangle$ is expected to overtake the other linear scores by at least a, i.e. (taking $a = 1$)

$$\langle w_y, x\rangle \geq 1 + \max_{k \neq y} \langle w_k, x\rangle \tag{6}$$

  and a corresponding relative multiclass *hinge loss* is:

$$l(x, W, y) = \left[1 - \langle w_y, x\rangle + \max_{k \neq y} \langle w_k, x\rangle\right]_+ \tag{7}$$

  with $[u]_+$ equal to u if $u \geq 0$ and 0 elsewhere.

- Compliant with the one-vs-all (OVA) construction (see also Allwein et al. (2000)), the *normative* margin setup imposes the classifier to provide a response that overtakes in norm a reference value a. Taking $a = 1$ for reference, it tells $\forall k$:

$$\langle w_k, x\rangle \geq 1 \qquad \text{if } y = k \tag{8}$$

$$\langle w_k, x\rangle \leq -1 \qquad \text{if } y \neq k \tag{9}$$

and a corresponding normative multiclass *hinge loss* is:

$$l(x, W, y) = \sum_{k=1}^{K} [1 + (1 - 2\delta(y, k))\langle w_k, x\rangle]_+ \tag{10}$$

The normative setup puts additional constraints on the classification task (see Crammer and Singer (2003)), but in counterpart provides mapping-independence across the different classes.

The hinge loss function comes with an implicit set point, namely $l_t = 0$, that grants response correctness under a margin constraint. In Kivinen et al. (2004), a loss-minimizer gradient descent update is combined with a norm-minimizer on the decision function, while Crammer et al. (2006) adopt a quadratic norm-minimal condition $\min_W ||W - W_{t-1}||^2$ on each update. In both cases local changes are shown to provide global improvement, e.g. Kivinen et al. (2004) show the stochastic gradient to converge to a global minimum and Crammer et al. (2006) show the total number of updates to be bounded in the linearly-separable case.

3. Our approach

3.1. Local quadratic optimization

On contrary to their supervised counterpart, class-separatrices updates in the Banditron (Kakade et al. (2008)) and PAB (Zhong et al. (2015)) are dense over time, loosing the Kernel-extension capability (at reasonable computational cost). The ability to store only the most relevant observation vectors is however a critical property of the discriminant approach we try to preserve in the bandit setup considered here. Adapting the one-vs-all normative margin constraint (presented in eqs.(8-9)) to the bandit case implies consider now the loss function:

$$\begin{aligned} l_t &= l(x_t, W_{t-1}, y_t, \tilde{y}_t) \\ &= [1 + (1 - 2\delta(y_t, \tilde{y}_t))\langle W_{t-1}, X_t^{\tilde{y}_t}\rangle]_+ \end{aligned} \tag{11}$$

with $\tilde{y}_t \in \{1, ..., K\}$ the single label to be compared with y_t, or :

$$l_t = [1 - \langle W_{t-1}, X_t^{\tilde{y}_t}\rangle]_+ \text{ if } y_t = \tilde{y}_t \tag{12}$$

$$l_t = [1 + \langle W_{t-1}, X_t^{\tilde{y}_t}\rangle]_+ \text{ elsewhere} \tag{13}$$

Algorithm 1 Bandit Passive-Aggressive (BPA)

Parameters: ε, C
Set $W \leftarrow \vec{0}$
for t in $[1, \ldots, T]$ **do**
  Read x_t
  Choose $\tilde{y}_t$
  Read $f_t = \delta(y_t, \tilde{y}_t)$
  $l_t \leftarrow \left[1 + (1 - 2f_t)\langle W, X_t^{\tilde{y}_t}\rangle\right]_+$
  $W \leftarrow W + \frac{l_t}{\|x_t\|^2 + \frac{1}{2C}}(2f_t - 1)X_t^{\tilde{y}_t}$
end for

In Crammer et al. (2006), a quadratic update norm minimization objective under a class-accuracy linear constraint l_t is considered. The weight update is the solution of :

$$W_t = \arg\min_W \frac{1}{2}\|W - W_{t-1}\|^2 + C\xi^2 \text{ s.t. } l_t \leq \xi$$

where C is an optional misclassification stiffness parameter. It provides here the following update :

$$W_t = W_{t-1} + \frac{l_t}{\|x_t\|^2 + \frac{1}{2C}}(2\delta(y_t, \tilde{y}_t) - 1)X_t^{\tilde{y}_t}$$

which leads to algorithm 1.

This setup is called "passive-aggressive" for it combines a conservative approach (ignore x_t if $l_t = 0$) with a tight update (optimize the classifier according to x_t if $l_t > 0$). In its original formulation ($C \to \infty$), this learning setup implements a "one-shot" update, i.e. carries out a zero-loss after update. A finite stiffness parameter C provides a more progressive (less "aggressive") update, allowing to deal more smoothly with outliers at the cost of a lesser sparsity.

Solving the system carries out in our case a single update of the $\tilde{y}_t^{\text{th}}$ separatrix. A rapid inspection shows that the full label information set could be used in the label hit case ($y = \tilde{y}$), allowing for K separatrices updates instead of one. This primary intuition is however not considered here, for multiple updates may put a too strong momentum on label hits. We show in the following that our careful and conservative approach is enough to provide strong convergence guarantees in most cases.

3.2. Linear separability

The passive-aggressive setup provides solid error bounds in the linearly-separable case. When trying to upper-bound the number of mistakes, it is worth considering an alternate classifier U that provides an alternate feedback $l_t^* = l(x_t, U, y_t, \tilde{y}_t)$. By construction, if the data vectors are separable under OVA constraints (see eq. (8-9)), there exist at least one classifier U such that $\forall(t, \tilde{y}_t), l_t^* = 0$. In that case, the following theorem holds:

Theorem 1. *Let $(x_1, y_1), ..., (x_T, y_T)$ be a sequence of separable examples where $x_t \in \mathbb{R}^d$, $y_t \in \{1, ..., K\}$ and $\| x_t \| \leqslant R$ for all t, let $\tilde{y}_1, ..., \tilde{y}_T$ be a sequence of responses, with $\tilde{y}_t \in \{1, ..., K\}$, and let $U \in \mathbb{R}^{Kd}$ be such that $\forall t, l_t^* = 0$. Then, assuming $C \to \infty$, the cumulative squared loss of algorithm 1 is bounded by:*

$$\sum_{t=1}^{T} l_t^2 \leqslant R^2 \| U \|^2 \tag{14}$$

(proof in Appendix A)

The result obtained in that case is formally similar, and even slightly tighter[2], than the one obtained by Crammer et al. (2006) in the multiclass case. Note however that the reference classifier U being defined in $\mathbb{R}^{Kd}$, while observation vectors are in $\mathbb{R}^d$, a linear dependence on the label set cardinality K is to be expected.

This result states, in short, that a finite number of updates is needed to fit the classification constraints expressed by the observed series of losses. In particular, for large series $(\tilde{y}_1, ..., \tilde{y}_t, ...)$, there is a point t^* at which all subsequent observed losses are equal to zero. This result grants the classifier finite complexity in the separable case whatever the number of samples.

There is however an important caveat to be mentioned. Indeed, given the $(\tilde{y}_1, ..., \tilde{y}_T)$ sequence, only T feedbacks signals out of KT potential feedbacks are actually read, and each separatrix w_k relies on a sub-series of observations $\{t : \tilde{y}_t = k\}$. The sequence probes the environment without necessarily uncovering all of it. The theorem provides a bound on the *observed* cumulative loss, ignoring every unobserved losses. Consequently, the loss (or squared loss) *is not an upper bound of the classification mistake.* The theorem thus does not guarantees that every example will be correctly classified in the

[2] using a relative hinge-loss, the multiclass upper bound is $4R^2 \| U \|^2$.

end. This would depends on (i) the particular policy followed in the course of learning and (ii) additional assumptions on sample regularity.

Let us now consider that a fixed policy is applied throughout the session, and let us assume that *every sample x from class k lies in a convex set $\mathcal{C}_k$* (observation sets convexity). Considering the theorem grants a zero loss after a fixed number of updates, let us note W^* this zero-loss final classifier and t^* the date of the final update. Then $\forall k \in 1, ..., K$,

Greedy deterministic policy. If $\tilde{y}_1, ..., \tilde{y}_t, ...$ obey to a greedy deterministic choice, i.e.

$$\forall t \in \{1, ..., T\}, \tilde{y}_t = h(x_t) = \underset{l \in \{1,...,K\}}{\operatorname{argmax}} \langle w_{l,t}, x_t \rangle$$

and if $\exists t \geq t^*$ such that $\tilde{y}_t = y_t = k$, then, as l_t is satisfied by W^*, $\langle w_k^*, x_t \rangle \geq 1$. Then, if $\exists x \in \mathcal{C}_k$ with $h(x) \neq k$, the zero-loss constraint implies that $\langle w_k^*, x \rangle < \langle w_{h(x)}^*, x \rangle \leq -1$. Then, following the convexity assumption, $\exists \rho \in [0, 1]$ such that $x_\rho = \rho x + (1 - \rho) x_t \in \mathcal{C}_k$ with $-1 < \langle w_k^*, x_\rho \rangle < 1$, so that $l(x_\rho, W^*, k, h(x_\rho)) \neq 0$ in any case, which breaks the zero loss condition. So, $\nexists x \in \mathcal{C}_k$ with $h(x) \neq k$, i.e. every sample from $\mathcal{C}_k$ is correctly classified.

Random uniform policy. If $\tilde{y}_1, ..., \tilde{y}_T$ obey to a uniform random choice (independent from W), i.e.

$$\forall t \in \{1, ..., T\}, P(\tilde{Y}_t = l) = \frac{1}{K}$$

the separatrix w_k is probed on average T/K times. By a simple combinatorial argument, the chance not finding $t > t^*$ such that $\tilde{y}_t = y_t = k$ exponentially decreases with $t - t^*$.

ε-greedy policy. Finally, an ε-greedy choice, i.e.

$$\forall t \in \{1, ..., T\}, P(\tilde{Y}_t = l) = (1 - \varepsilon)\delta(l, h(x_t)) + \frac{\varepsilon}{K}$$

with a (fixed) exploration parameter $\varepsilon \in [0, 1]$ alternates between uniform sampling and greedy choice. Then, finding $t > t^*$ such that $\tilde{y}_t = y_t = k$ is provided almost surely by the uniform sampling, while the greedy choice combined with the convexity assumption leverages the correct classification of every sample belonging to class k.

We consistently adopt an ε-greedy approach in simulations, i.e. comply with the Kakade et al. (2008) formula.

3.3. Stationary case

If we turn now to an arbitrary classifier U, i.e. do not take for granted the separability assumption, then, for any given dataset, the following theorem holds :

Theorem 2. *Let $(x_1, y_1), ..., (x_T, y_T)$ be a sequence of examples where $x_t \in \mathbb{R}^d$, $y_t \in \{1, ..., K\}$ and $\| x_t \| \leqslant R$ for all t, let $\tilde{y}_1, ..., \tilde{y}_T$ be a sequence of responses, with $\tilde{y}_t \in \{1, ..., K\}$. Then for any $U \in \mathbb{R}^{Kd}$, and assuming $C \to \infty$, the cumulative squared loss of algorithm 1 is bounded by:*

$$\sum_{t=1}^{T} l_t^2 \leqslant \left(R \parallel U \parallel + 2\sqrt{\sum_{t=1}^{T} (l_t^*)^2} \right)^2$$

(proof in Appendix B)

This bound is once again similar to that obtained by Crammer et al. (2006) in the supervised case. It tells in short that, for large T, the average squared loss will be in the worst case four time that of any linear classifier U (including that of a loss-minimizer classifier U^*). In approximation[3], the final loss is thus expected to be on average twice that of the best classifier. As shown in Crammer et al. (2006)), this error bound is less tight than that of Freund and Schapire (1997), and additional regularization (with finite C) is needed to attain $O(\sqrt{T})$ regret, we do not develop here for brevity.

Given the aggressiveness of the algorithm, this "loss-doubling" can be interpreted the following way: in the long run, each non-zero loss encountered is expected to provoke an aggressive (overfitting) update, that will need on average a same amount of loss in subsequent rounds to recover. When the proportion of outliers is not to strong, the separability assumption can be seen as a boundary condition to which the classifier periodically returns between temporary excursions throughout disrupting updates and recovery.

Like previously, this result is not a strict bound on the classification error, for a mere sample of the total OVA losses is visited in one run. The separable case is now the boundary condition toward which the classifier continually returns, for which specific policies need to be assumed to provide effective final classification (see previous section).

3.4. Gradient-descent approach

[3]If the loss variance is small, i.e. $\langle l_t^2 \rangle_t \simeq \langle l_t \rangle_t^2$.

Algorithm 2 H-horizon Stochastic Gradient Descent (SGD)

Parameters: ε, η, λ, H
Set $W \leftarrow \vec{0}$, $n \leftarrow 0$, $b_{\text{inf}} \leftarrow 1$
for t in $[1, \ldots, T]$ **do**
  Read x_t
  Choose $\tilde{y}_t$
  Read $f_t = \delta(y_t, \tilde{y}_t)$
  $l_t \leftarrow \left[1 + (1 - 2f_t)\langle W, X_t^{\tilde{y}_t}\rangle\right]_+$
  if $l_t > 0$ **then**
    $n \leftarrow n + 1$
    $\alpha_n \leftarrow \eta(2f_t - 1)$
    Store $X_n = X_t^{\tilde{y}_t}$
    if $n > H$ **then**
      $b_{\text{inf}} \leftarrow n - H + 1$
      Erase X_{n-H} from memory
    end if
    for i in $[b_{\text{inf}}, ..., n-1]$ **do**
      $\alpha_i \leftarrow (1 - \eta\lambda)\alpha_i$
    end for
    $W \leftarrow \sum_{i=b_{\text{inf}}}^{n} \alpha_i X_i$
  end if
end for

Aside the quadratic optimization setup presented above, the gradient-based approach to sparse online discriminative classification, as proposed by Kivinen et al. (2004), relies on minimizing the regularized risk:

$$R(W) = \mathbb{E}\left[l + \frac{\lambda}{2}\|W\|^2\right]$$

with λ a regularization parameter.

Taking l_t as in eq.(11), the regularized risk gradient estimator at each step t can be shown to be:

$$g_t = \begin{cases} \lambda W_{t-1} + (1 - 2\delta(y_t, \tilde{y}_t))X_t^{\tilde{y}_t} & \text{if } l_t > 0 \\ \lambda W_{t-1} & \text{elsewhere } (l_t = 0) \end{cases}$$

and a stochastic gradient descent approach with learning parameter η pro-

vides the following update:

$$W_t = \begin{cases} (1-\eta\lambda)W_{t-1} - \eta(1-2\delta(y_t,\tilde{y}_t))X_t^{\tilde{y}_t} & \text{if } l_t > 0 \\ (1-\eta\lambda)W_{t-1} & \text{elsewhere } (l_t = 0) \end{cases}$$

In practice, the classifier output remaining unchanged when $l_t = 0$, and, following a conservative approach, we omit the update in that case.

Then, the classifier can be made explicit in the form of a sum over observation vectors :

$$W_t = \sum_{t'=1}^{t} \alpha_{t'} X_t^{\tilde{y}_{t'}}$$

with:

$$\alpha_{t'} = \mathbf{1}_{\{l_{t'}>0\}}(1-\eta\lambda)^{\sigma_t - \sigma_{t'} - 1}\eta(2\delta(y_{t'},\tilde{y}_{t'}) - 1)$$

with $\mathbf{1}_{\{u\}}$ equal to 1 when u is true and 0 elsewhere, and σ_t the number of updates at time t, i.e.

$$\sigma_t = \sum_{t'=1}^{t} \mathbf{1}_{\{l_{t'}>0\}}$$

and the cardinality of the non-zero coefficients correspond to the number of observation vectors effectively stored in memory.

Then, following Kivinen et al. (2004), a strict control on the number of prototype vectors may be imposed, though "old" coefficients exponentially vanish while new updates take place. A H-horizon truncation principle may be adopted, with every $\alpha_{t'}$ such that $\sigma_t - \sigma_{t'} > H$ set to 0. The truncation error can then be shown to exponentially decrease with H (see Kivinen et al. (2004)). This approach is of course well-adapted to the non-stationary case, where context-related categories change over time.

3.5. Kernel Extension

The extension of the linear discriminant setup to RKHS (Reproducible Kernel Hilbert Space) redescription spaces allows to deal with non-linearly separable learning sets at the cost of additional free parameters (kernel specific parameters).

Let $\mathcal{K}(.,.)$ a mapping from $\mathbb{R}^d \times \mathbb{R}^d$ to $\mathbb{R}^+$ having the reproducing property (see Schölkopf and Smola (2002)). Let $\mathcal{K}(x,.)$ be the projection of example x in $\mathcal{H}$. Then, by construction, a scalar product $\langle .,. \rangle_{\mathcal{H}}$ in $\mathcal{H}$ is such that:

$$\forall w \in \mathcal{H}, \langle w, \mathcal{K}(x,.)\rangle_{\mathcal{H}} = w(x)$$

Keeping previous notations, each classifier $W = (w_1, ..., w_k)$ is now defined in $\mathcal{H}^K$ and:

$$X^k \triangleq (0(.), ..., \underset{|}{\underset{k}{\mathcal{K}(x, .)}}, ..., 0(.)) \in \mathcal{H}^K \tag{15}$$

with $0(.)$ a null function. Then, taking:

$$\langle W, X^k \rangle \triangleq \langle w_k, \mathcal{K}(x, .) \rangle_{\mathcal{H}} = w_k(x)$$

all previous definitions an results apply in the redescription space.

In particular, considering algorithm 1 and noting $\alpha_t = \frac{l_t}{\mathcal{K}(x_t, x_t) + \frac{1}{2C}}(2f_t - 1)$, we have:

$$W_t = \sum_{t'=1}^{t} \alpha_{t'} X_{t'}^{\tilde{y}_{t'}}$$

so that, taking definition (15), and setting $\forall k, \mathcal{T}_{k,t} = \{t' \leq t : \tilde{y}_{t'} = k \text{ and } l_{t'} > 0\}$, each separatrix $w_{k,t}$ is now defined by a set of prototypes $\{x_{t'}\}_{t' \in \mathcal{T}_{k,t}}$ so that

$$w_{k,t} = \sum_{t' \in \mathcal{T}_{k,t}} \alpha_{t'} \mathcal{K}(x_{t'}, .)$$

The total number of prototypes is incremented each time a non-zero loss is read out. From theorem 1, we know this number is bounded since every dataset is separable in infinite-dimensional redescription spaces.

4. Experiments

4.1. Datasets

Different algorithms are evaluated on two synthetic and three real-world datasets. Their principal characteristics are provided in table 1.

Under the contextual bandit setup, most numerical experiments found in literature concentrate on text-mining applications (see for instance Crammer and Gentile (2013)), having both large dimensional sparse vector text representations, a large number of examples, and up to 30-50 labels. Such databases provide a good testbed for realistic scale constraints, with linear models generally effective to separate the data. In order to both test scaling

Table 1: Five datasets considered, with n the number of instances, d the vectors dimension and K the number of labels.

Dataset	n	d	K
SynSep	10^5	400	9
SynNonSep	10^5	400	9
RCV1-v2	10^5	47236	53
Segment	2310	19	7
Pendigits	7494	16	10

and sparsity, we here both consider text mining databases and more traditional machine learning databases, having smaller memory footprint but stronger non-linear constraints.

Our two first datasets mimick text documents vectors, with small 400-dimensional vocabulary. 10^5 instances, belonging to 9 different classes, are generated. The detailed construction of those datasets is given in Kakade et al. (2008). In the first one, called *SynSep*, the different classes are linearly separable. In the second one, called *SynNonSep*, a 5% label noise is introduced, rendering the dataset non separable.

The third dataset comes from the Reuters *RCV1-v2* collection (see Lewis et al. (2004)). This dataset is a typical text mining setup, containing both high dimensional vectors (47,236 vocabulary entries) and a large number of instances (10^5). The original dataset contains multi-label instances. In order to fit the single label setup, we adopt the preprocessing method proposed by Bekkerman and Scholz (2008), issuing a 53-class dataset.

The two last datasets are typical machine learning real-world datasets, having a smaller number of instances and a smaller dimension, allowing to test sparsity in the Kernel embedding case. The fourth dataset, named *Segment* (UCI's Image Segmentation Data Set – Lichman (2013)), owns 2310 instances. Each feature vector is build from 3×3 pixels excerpts from natural images, with 19 features per instance and 7 classes. The fifth dataset, named *Pendigits* (UCI's Pen-Based Recognition of Handwritten Digits Data Set – Alimoglu et al. (1996)), is based on the preprocessed (normalization and downsampling) of (x, y) coordinate encoded handwritten digits. It owns 7494 instances, with 16 features per instance and 10 classes.

4.2. Algorithms

Table 2: Parameters setting for different algorithms and different datasets. **P** stands for Perceptron, **PA** for Passive Agressive, **B** for Banditron, **C** for Confidit, **BPA** for the Bandit Passive Aggressive (algorithm 1), **K-B** for the kernel Banditron, **K-BPA** for the kernel realization of BPA (algorithm 1) and **K-SGD** for the kernel realization of SGD (algorithm 2).

Dataset	**P**	**PA**	**B**	**C**	**BPA**
Synsep	$\varnothing$	$C \to \infty$	$\varepsilon = 0.014$	$\eta = 10^3$	$\varepsilon = 0.4$ $C \to \infty$
SynNonSep	$\varnothing$	$C = 10^{-2}$	$\varepsilon = 0.65$	$\eta = 10^3$	$\varepsilon = 0.8$ $C = 10^{-2}$
RCV1-v2	$\varnothing$	$C = 10^{-2}$	$\varepsilon = 0.4$	$\eta = 10^2$	$\varepsilon = 0.2$ $C = 10^{-2}$
	K-B	**BPA**	**K-BPA**	**K-SGD**	
Segment	$\sigma = 1$ $\varepsilon = 0.1$	$\varepsilon = 0.3$	$\sigma = 1$ $\varepsilon = 0.3$	$\sigma = 1$ $H = 200$	
Pendigits	$\sigma = 10$ $\varepsilon = 0.1$	$\varepsilon = 0.3$	$\sigma = 10$ $\varepsilon = 0.3$	$\sigma = 10$ $H = 500$	

Only online learning methods are here considered for comparison. For a given dataset, each instance is presented once. The classifier update starts right after the first instance presentation, and finishes at the last one. For each instance, a single response is carried out, and a single corresponding feedback is obtained.

For comparison, both full-feedback and one-bit feedback learning setups are tested:

- The multiclass perceptron (see Duda et al. (1973)) and the multiclass passive-aggressive setup (see Crammer et al. (2006)) have a full feedback at disposal.
- The Banditron (Kakade et al. (2008)), Confidit (Crammer and Gentile (2013)), and our algorithms BPA (algorithm 1) and HGD (algorithm 2) only have a one-bit feedback at disposal.

In the specific kernel case (Segment and Pendigits databases), the non-separability assumption allows to withdraw the stiffness parameter, i.e. to

consider $C \to \infty$. The Kernel function used in simulations is the Laplacian kernel, i.e. :

$$K(x, y) = \exp\left(-\frac{\| x - y \|}{\sigma}\right)$$

whose radius is set by parameter $\sigma > 0$. The Confidit Algorithm, for which the kernel extension is not straightforward, is not tested in that case.

In order to faithfully compare the methods, the different parameters (if any) are calculated by cross-validation over the final classification rate. The resulting parameters, as used in simulations, are given in table 2.

4.3. Metrics

The metrics used to compare the different algorithms are the cumulative error rate and the average error rate.

- The cumulative error rate is defined at each round t as the sum of errors until t, i.e.:

$$M_t = \sum_{t'=1}^{t} \mathbf{1}_{\hat{y}_{t'} \neq y_{t'}}$$

  with $\hat{y}_t$ as defined in eq. (2).

- The average error rate, carried out over 100-round sliding windows, allows for a more refined estimate of the continuing improvement over learning sessions, i.e.:

$$\forall t > 99, \bar{m}_t = \frac{1}{100} \sum_{t'=t-99}^{t} \mathbf{1}_{\hat{y}_{t'} \neq y_{t'}}$$

Remark. Both the Banditron and our bandit reduction to the OVA Passive-Aggressive setup (Algorithm 1) use an ε-greedy exploration policy, that has direct effect on the classification rate, for the actual response $\tilde{y}_t$ differs from $\hat{y}_t$ in a proportion equal to ε. This residual exploration error persists whatever effective the classifier W is at separating the data. In order to properly compare algorithms, we need to evenly evaluate improvement across ε values, i.e. only use the internal noise-free estimate $\hat{y}_t$ for evaluation.

Under the kernel approach, the number of prototype vectors is expected to grow up across the learning session, making the response delay grow in proportion. The actual calculation time c_t is measured at each round during

learning sessions, and an average over 100 ms sliding windows is calculated as follows:

$$\forall t > 99, \bar{c}_t = \frac{1}{100} \sum_{t'=t-99}^{t} c_{t'}$$

4.4. Results

We investigate on Figure 1 the effect of the exploration parameter ε on the final classification rate, for the Banditron and the Bandit Passive-Aggressive (algorithm 1). Apart from better final classification rate, the passive aggressive setup is shown insensitive to variable ε, on contrary to the Banditron having a known exploration rate dependence. Greedy exploitation and pure exploration show similar effectiveness in our case. In practice, this result allows consider decreasing exploration rates over learning sessions, for to take advantage of the classification rates attained in the course of learning.

The Perceptron, Passive-Aggressive (PA), Banditron, Confidit and Bandit Passive Aggressive (BPA) cumulative errors are compared on figure 2 over the SynSep, SynNonSep and Reuters RCV1-v2 datasets.

The first dataset being linearly separable, a final error bound is rapidly attained by all methods except for the Banditron showing only a monotonic decrease of the error rate. The convergence is shown even faster for the supervised methods (Perceptron and PA), for barely 20 errors are observed over 10^5 instances.

In the SynNonSep case, as the 5% label noise is irreductible, all methods show a linear growth of the cumulative error. However, the slower increase seen on Confidit and BPA point out a better resilience to label noise, when compared to the supervised setup. This resilience, that was noticed in Crammer and Gentile (2013) and Ngo et al. (2013), thus extends here to the BPA approach.

The Reuters RCV1-v2 database provides a scale-realistic testbed. The sub-linear cumulative error rate shown by all methods indicate that learning is effective in all case, with, however, a clear gap between the Banditron and the other methods. In detail, the two supervised algorithms outperform by little BPA, followed by Confidit, and then the Banditron far behind. The first four methods show almost similar final slopes, on contrary to the Banditron showing a higher final error rate (see figure 1). The good performances of BPA and Confidit are noticeable here, for the number of classes is high (53) and the labelling information consequently very scarce. Moreover, the

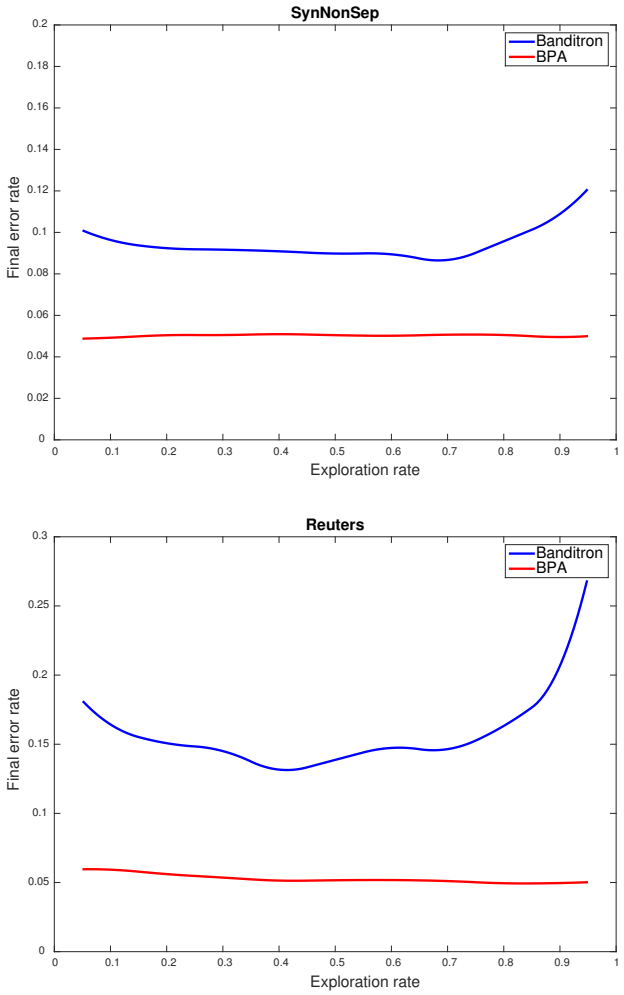

Figure 1: Banditron and Bandit Passive Aggressive (BPA) final error rate in function of the exploration rate ε, on SynNonSep (top) and Reuters (bottom) datasets. Parameters are in table 2.

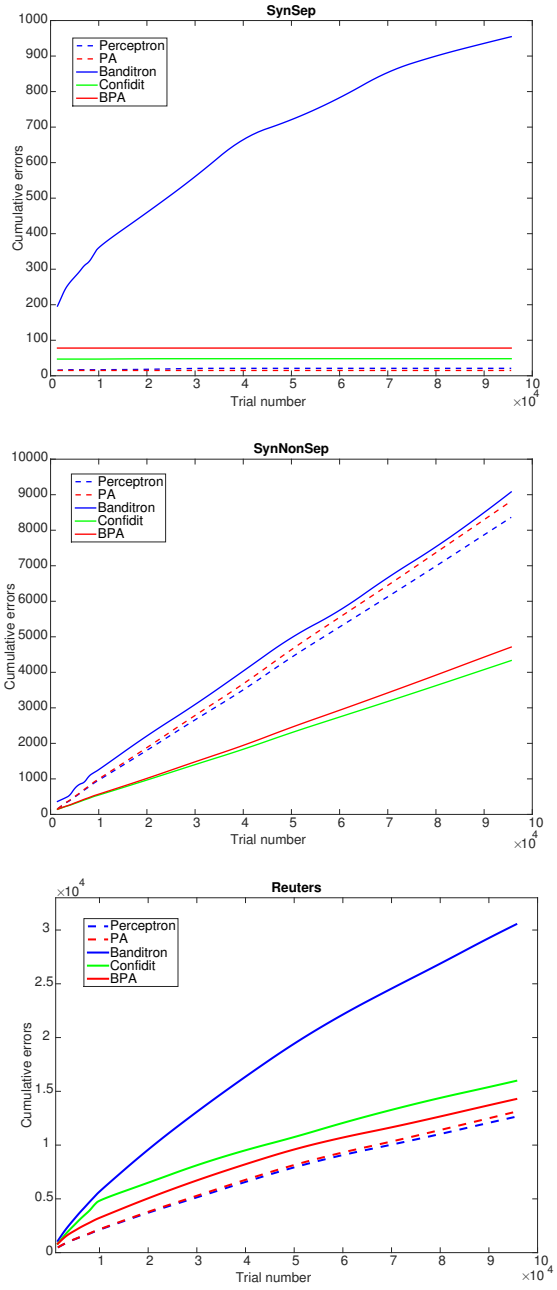

Figure 2: Perceptron, Passive-Aggressive (PA), Banditron, Confidit and Bandit Passive Aggressive (BPA) cumulative errors over trial number for the SynSep (9 classes), SynNonSep (9 classes) and Reuters (53 classes) databases. Parameters are in table 2.

marked prevailing of BPA over Confidit is also noticeable for its algorithmic complexity is much less[4].

The Segment and Pendigits datasets, to which Linear BPA, kernel BPA, kernel SGD and kernel Banditron are applied on figures 3 and 4, present a dramatic decrease in size, with vectors of only 19 features in Segment and only 16 in Pendigits. These more reduced feature spaces are counterbalanced by a stronger non-separability, that dictates the use of kernel methods. Online learning with kernels being generally burdened by the increasing size of the prototype vectors set (see Kivinen et al. (2004)), we check here for the sparsity of the different setups. In particular, we compare the K-BPA native sparsity (algorithm 1) with the explicit sparsity control of the SGD method (algorithm 2), while the linear BPA provides a baseline reference, and the kernel-Banditron illustrates the upper bound computational cost of a non-sparse update.

The increasing computational cost over time is shown on figure 3. Apart from the linear BPA baseline constant cost, all kernel-based setups show a monotonic increase over time, with the most parsimonious trend obtained by the K-SGD, followed by the K-BPA, and then the K-Banditron constant trend. The computational cost of a learning session is consequently $0(T)$ for the linear BPA (best case), $O(T^2)$ for the K-Banditron (worst case) and in between for the two other cases. The BPA algorithm is expected to reach a constant cost after the final number of prototypes is reached, this number being set to 200 on the Segment database and 500 on the Pendigits one. A plateau is roughly observed after 1000 trials on the Segment database, while a continuing complexity rise is observed at slow rate in the other case. The K-BPA, while still more costly, shows a very close-by trend on the Segment database, and a more marked difference on the Pendigits database. These results confirm in general the sparsity effectiveness, and the subsequent reduced computational cost, of the K-BPA setup.

Now turning to the classification rates, the cumulative errors obtained by the linear BPA, kernel BPA, kernel SGD and kernel Banditron on the Segment and Pendigits databases are shown on figure 4. The first thing to be noticed is the prevailing of the Kernel setups over the linear one in both cases, and particularly on the Segment database where the linear BPA

[4]Confidit uses a second order sample covariance estimate, this covariance matrix being reduced to a mere diagonal in the high-dimensional case (see Crammer and Gentile (2013)).

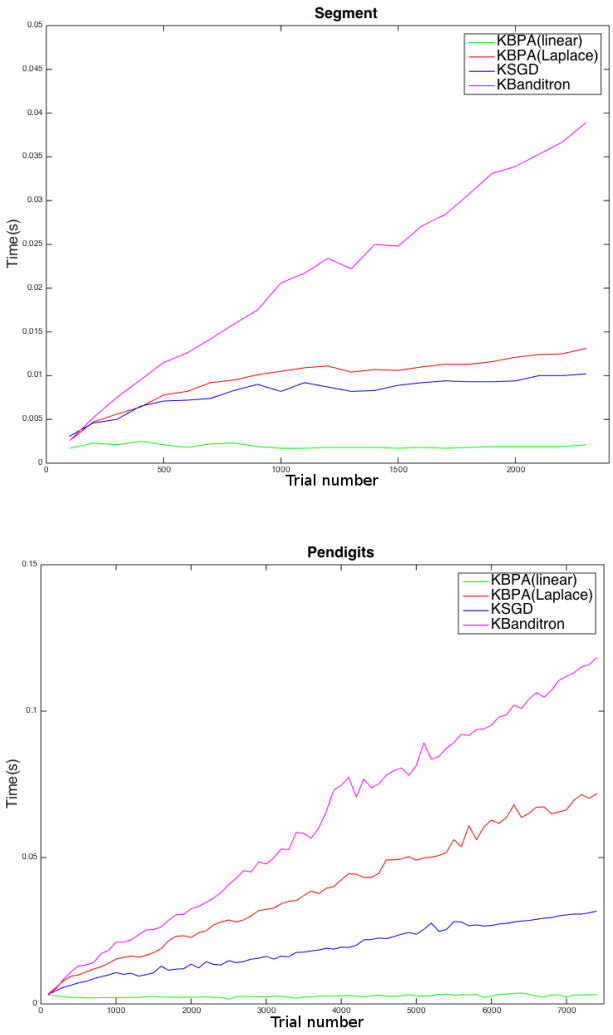

Figure 3: Linear BPA (algorithm 1), kernel BPA (algorithm 1), kernel SGD (algorithm 2) and kernel Banditron average computational cost over trial number on Segment (top) and Pendigits (bottom) databases. Parameters are in table 2.

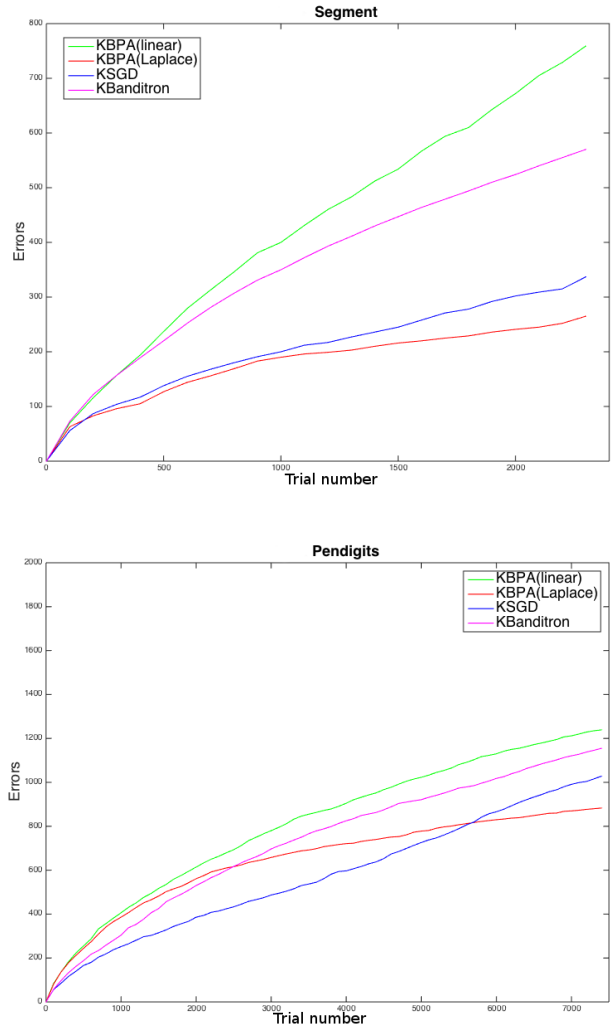

Figure 4: Linear BPA (algorithm 1), kernel BPA (algorithm 1), kernel SGD (algorithm 2) and kernel Banditron cumulative errors over trial number on Segment (top) and Pendigits(bottom) databases. Parameters are in table 2.

hardly shows improvement over time. The banditron is generally close to this worst-case scenario, while the K-SGD, despite a good initial startup, experience difficulty to capitalize improvement over time (more particularly on the Pendigits case). In contrast, the kernel BPA is clearly found to outperform the other methods, in particular regarding the final error rate: the final less than 2% error rate obtained on the Pendigits database (not shown) overtakes the other methods, but also approaches the state-of-the-art classification rates obtained in offline/full feedback settings.

5. Conclusion

We have shown that a conservative reduction of the OVA hinge-loss provides an effective and lightweight solutions to the bandit classification problem (as defined by Kakade et al. (2008)). In particular, when adapting the passive aggressive setup proposed in the supervised case by Crammer et al. (2006), we prove similar bounds on the observed cumulative squared loss. Here, however, the bound is not an upper bound of the classification error. Additional regularity assumptions, such as observation sets convexity, or a partly uniform sampling of the label space, need to be considered to provide comparable upper error bounds. In addition, as in Crammer et al. (2006), a soft margin stiffness parameter C needs to be optimized to reach $O(\sqrt{T})$ regret in the stationary case.

The numerical simulations provide favorable results on both large scale text mining datasets and non-linearly separable machine learning datasets. When comparing our approach with the Banditron, a first result is the exploration parameter ε insensitivity, allowing to consider ε decrease over time in practical implementations. When comparing with confidence-based second-order contextual bandit approaches (see Crammer and Gentile (2013)), our approach also shows favorable results on a scale-realistic dataset, while owning a much lesser complexity. A good resilience to label noise is also shown on a synthetic dataset, for the passive-aggressive method encompass a soft margin principle allowing to efficiently deal with outliers.

In complement, a kernel approach was implemented on non-linearly separable datasets, and shown effective when both considering sparsity (such as addressed by Kivinen et al. (2004)), and final classification accuracy, with, for instance, a close to state-of-the-art 98% final accuracy observed after one pass on the Pendigits 10-class problem.

In conclusion, our approach provides many practical advantages when considering the bandit classification problem, among which (i) simplicity, with easy algorithmic implementation, (ii) linear scaling in space, (iii) sparsity, (iv) linear scaling to the labels space cardinality, (v) kernel compatibility and (vi) resilience to label noise. It moreover reveals surprisingly effective when compared to the more elaborate noise-aware UCB-like setups. This is, to our best knowledge, the first adaptation of sparse online learning principles to the bandit case, providing avenue toward effective kernel implementations of non-linearly separable contextual bandit setups. Our OVA reduction may even generalize to slightly more demanding tasks, like the multi-label setup generally considered in recommender systems. Complement investigations are however needed to address both the non-stationary and adversarial cases. The H-horizon window approach, as proposed by Kivinen et al. (2004), may in some cases be substituted, for it shows both effectiveness, sparsity and adaptivity in simulations.

From a more general standpoint, the bandit classification problem implements a form of active learning in scarcely labeled environments, with a limited (1 bit) information budget at each round. Apart for text mining and recommender systems, it would probably need additional developments to reach full relevance in real-world problems having both a sequential organization and undergoing a strict "win-or-loose" return. In the case of artificial games, for instance, both multiple moves and temporal credit assignment (see Sutton and Barto (1998)) would need to be considered in addition. Problems implying non-vectorial spaces (like graph spaces) and corresponding similarity metrics may also be addressed, just like it is the case for other margin-based risk-minimizer techniques (see Chen et al. (2009)).

Acknowledgements

This work was supported by the China Studentship Council (CSC). Thanks to Liva Ralaivola (Aix Marseille Univ, LIF, Marseille, France) for fruitful discussions.

Appendix A. Proof of Theorem 1

Proof. Define Δ_t to be:

$$\Delta_t = \| W_{t-1} - U \|^2 - \| W_t - U \|^2$$

Summing Δ_t over all t from 1 to T collapses to:

$$\begin{aligned}\sum_{t=1}^{T}\Delta_t &= \sum_{t=1}^{T}\left(\parallel W_{t-1}-U\parallel^2 - \parallel W_t-U\parallel^2\right)\\ &= \parallel W_0-U\parallel^2 - \parallel W_T-U\parallel^2\end{aligned}$$

Given that $W_0=\vec{0}$,

$$\sum_{t=1}^{T}\Delta_t = \parallel U\parallel^2 - \parallel W_T-U\parallel^2 \leqslant \parallel U\parallel^2 \tag{A.1}$$

Using the definition of update :

$$\Delta_t = -2\left\langle W_{t-1}-U, (2f_t-1)\frac{l_t}{\parallel x_t\parallel^2}X_t^{\tilde{y}_t}\right\rangle - \left\|\frac{l_t}{\parallel x_t\parallel^2}X_t^{\tilde{y}_t}\right\|^2$$

So, taking $\parallel X_t^{\tilde{y}_t}\parallel = \parallel x_t\parallel$, it comes:

$$\Delta_t = 2l_t\frac{(1-2f_t)\langle W_{t-1}, X_t^{\tilde{y}_t}\rangle - (1-2f_t)\langle U, X_t^{\tilde{y}_t}\rangle}{\|x_t\|^2} - \frac{l_t^2}{\parallel x_t\parallel^2}$$

Then, noting that:

$l_t = [1+(1-2f_t)\cdot\langle W_{t-1}, X_t^{\tilde{y}_t}\rangle]_+$

$l_t^* = [1+(1-2f_t)\cdot\langle U, X_t^{\tilde{y}_t}\rangle]_+$

that $\Delta_t = 0$ when $l_t = 0$, and that $l_t^* \geq 1+(1-2f_t)\cdot\langle U, X_t^{\tilde{y}_t}\rangle$, it comes :

$$\begin{aligned}\Delta_t &\geqslant 2l_t\frac{l_t-l_t^*}{\parallel x_t\parallel^2} - \frac{l_t^2}{\parallel x_t\parallel^2}\\ &= \frac{l_t^2-2l_tl_t^*}{\parallel x_t\parallel^2}\end{aligned}$$

Given that U is such that $\forall t\in[1,...,T]$, $l_t^*=0$,

$$\begin{aligned}\Rightarrow \parallel U\parallel^2 &\geqslant \sum_{t=1}^{T}\Delta_t \geqslant \sum_{t=1}^{T}\frac{l_t^2}{\parallel x_t\parallel^2} \geqslant \sum_{t=1}^{T}\frac{l_t^2}{R^2}\\ \Rightarrow \sum_{t=1}^{T}l_t^2 &\leqslant R^2\cdot\parallel U\parallel^2\end{aligned}$$

□

Appendix B. Proof of Theorem 2

Proof. From the proof of Theorem 1,

$$\sum_{t=1}^{T} l_t^2 \leqslant R^2 \cdot \parallel U \parallel^2 + 2 \sum_{t=1}^{T} l_t l_t^*$$

To upper bound the right side of the above inequality, we denote $a_t = \sqrt{\sum_{t=1}^{T} l_t^2}$ and $b_t = \sqrt{\sum_{t=1}^{T} (l_t^*)^2}$,

$$\begin{aligned} 2(a_t b_t)^2 - 2(\sum_{t=1}^{T} l_t l_t^*)^2 &= \sum_{i=1}^{T} \sum_{j=1}^{T} l_i^2 (l_j^*)^2 + \sum_{i=1}^{T} \sum_{j=1}^{T} l_j^2 (l_i^*)^2 \\ &\quad - 2 \sum_{i=1}^{T} \sum_{j=1}^{T} l_i l_j l_i^* l_j^* \\ &= \sum_{i=1}^{T} \sum_{j=1}^{T} (l_i l_j^* - l_j l_i^*)^2 \geqslant 0 \end{aligned}$$

$$\sum_{t=1}^{T} l_t^2 \leqslant R^2 \cdot \parallel U \parallel^2 + 2 \sum_{t=1}^{T} l_t l_t^* \leqslant R^2 \cdot \parallel U \parallel^2 + 2 a_t b_t$$

then considering:

$$a_t^2 - 2 a_t b_t + b_t^2 \leqslant R^2 \parallel U \parallel^2 + b_t^2$$

we obtain :

$$a_t \leqslant b_t + \sqrt{R^2 \parallel U \parallel^2 + b_t^2}$$

and using the fact that $\sqrt{a+b} \leqslant \sqrt{a} + \sqrt{b}$,

$$a_t \leqslant R \parallel U \parallel + 2 b_t$$

so that :

$$\sum_{t=1}^{T} l_t^2 \leqslant \left(R \parallel U \parallel + 2 \sqrt{\sum_{t=1}^{T} (l_t^*)^2} \right)^2$$

□

References

Alimoglu, F., Doc, D., Alpaydin, E., Denizhan, Y., 1996. Combining multiple classifiers for pen-based handwritten digit recognition.

Allwein, E. L., Schapire, R. E., Singer, Y., 2000. Reducing multiclass to binary: A unifying approach for margin classifiers. Journal of machine learning research 1 (Dec), 113–141.

Amari, S.-I., Park, H., Fukumizu, K., 2000. Adaptive method of realizing natural gradient learning for multilayer perceptrons. Neural Computation 12 (6), 1399–1409.

Anlauf, J., Biehl, M., 1989. The adatron: an adaptive perceptron algorithm. EPL (Europhysics Letters) 10 (7), 687.

Auer, P., 2002. Using confidence bounds for exploitation-exploration tradeoffs. Journal of Machine Learning Research 3 (Nov), 397–422.

Auer, P., Cesa-Bianchi, N., Fischer, P., 2002. Finite-time analysis of the multiarmed bandit problem. Machine learning 47 (2-3), 235–256.

Bekkerman, R., Scholz, M., 2008. Data weaving: Scaling up the state-of-the-art in data clustering. No. 1083–1092. In Proceedings of CIKM.

Cesa-Bianchi, N., Conconi, A., Gentile, C., 2005. A second-order perceptron algorithm. SIAM Journal on Computing 34 (3), 640–668.

Chen, Y., Garcia, E. K., Gupta, M. R., Rahimi, A., Cazzanti, L., 2009. Similarity-based classification: Concepts and algorithms. Journal of Machine Learning Research 10 (Mar), 747–776.

Crammer, K., Dekel, O., Keshet, J., Shalev-Shwartz, S., Singer, Y., 2006. Online passive-aggressive algorithms. The Journal of Machine Learning Research 7, 551–585.

Crammer, K., Gentile, C., 2013. Multiclass classification with bandit feedback using adaptive regularization. Machine learning 90 (3), 347–383.

Crammer, K., Singer, Y., 2003. Ultraconservative online algorithms for multiclass problems. The Journal of Machine Learning Research 3, 951–991.

Duda, R. O., Hart, P. E., et al., 1973. Pattern classification and scene analysis. Vol. 3. Wiley New York.

Freund, Y., Schapire, R. E., 1997. A decision-theoretic generalization of online learning and an application to boosting. Journal of computer and system sciences 55 (1), 119–139.

Freund, Y., Schapire, R. E., 1999. Large margin classification using the perceptron algorithm. Machine learning 37 (3), 277–296.

Hazan, E., Kale, S., 2011. Newtron: an efficient bandit algorithm for online multiclass prediction. In: Advances in Neural Information Processing Systems. pp. 891–899.

Kakade, S. M., Shalev-Shwartz, S., Tewari, A., 2008. Efficient bandit algorithms for online multiclass prediction. In: Proceedings of the 25th international conference on Machine learning. ACM, pp. 440–447.

Kivinen, J., Smola, A. J., Williamson, R. C., 2004. Online learning with kernels. Signal Processing, IEEE Transactions on 52 (8), 2165–2176.

Lai, T. L., Robbins, H., 1985. Asymptotically efficient adaptive allocation rules. Advances in applied mathematics 6 (1), 4–22.

Le Cun, L. B. Y., Bottou, L., 2004. Large scale online learning. Advances in neural information processing systems 16, 217.

Lewis, D., Yang, Y., Rose, T., Li, F., 2004. Rcv1: A new benchmark collection for text categorization research. Vol. 5 of JMLR. pp. 361–397.

Li, L., Chu, W., Langford, J., Schapire, R. E., 2010. A contextual-bandit approach to personalized news article recommendation. In: Proceedings of the 19th international conference on World wide web. ACM, pp. 661–670.

Lichman, M., 2013. UCI machine learning repository.
URL `http://archive.ics.uci.edu/ml`

Ngo, H. Q., Luciw, M. D., Vien, N. A., Schmidhuber, J., 2013. Upper confidence weighted learning for efficient exploration in multiclass prediction with binary feedback. In: IJCAI.

Robbins, H., 1952. Some aspects of the sequential design of experiments. Bulleting of the American Mathematical Society 58, 527–535.

Rosenblatt, F., 1958. The perceptron: a probabilistic model for information storage and organization in the brain. Psychological review 65 (6), 386.

Schölkopf, B., Smola, A. J., 2002. Learning with kernels: support vector machines, regularization, optimization, and beyond. MIT press.

Sutton, R. S., Barto, A. G., 1998. Reinforcement learning: An introduction. Vol. 1. MIT press Cambridge.

Vapnik, V. N., 1998. Statistical learning theory. Vol. 1. Wiley New York.

Zhong, H., Daucé, E., Ralaivola, L., 2015. Online multiclass learning with "bandit" feedback under a passive-aggressive approach. in proceedings of European Symposium on Artificial Neural Networks Computational Intelligence and Machine Learning (ESANN), pp. 403–408.

Zhong, H., Daucé, E., 2015. Passive-aggressive bounds in bandit feedback clasification. In: Hollmén, J., Papapetrou, P. (Eds.), proc. of the ECML-PKDD 2015 Doctoral Consortium. Aalto University, pp. 255– 264.